

Gregorics Tibor
EHACODE.ELTE
gt@inf.elte.hu
0.csoport

1. beadandó/0.feladat

2011. december 28.

Feladat

Egy osztályba n diák jár, akik m darab tantárgyat tanulnak. Ismerjük a félév végi osztályzataikat. Igaz-e, hogy minden diáknak van legalább két ötöse?

Megoldási terv

$A = (\text{napló}: \mathbb{N}^{n \times m}, l: \mathbb{L})$
 $Ef = (\text{napló} = \text{napló}')$

$Uf = (Ef \wedge (l = \forall i \in [1..n] : \text{ötösdb}(i) \geq 2)) = (Ef \wedge (l = \forall_{i=1}^n \text{search}(\text{ötösdb}(i) \geq 2)))$

ahol $\text{ötösdb}(i) : [1..n] \rightarrow \mathbb{N}$ és $\text{ötösdb}(i) = \sum_{j=1}^m 1_{\text{napló}[i,j]=5}$

$l, i := \text{true}, 1$	$s := \text{ötösdb}(i)$
$l \wedge i \leq n$	$s := 0$
$l := \text{ötösdb}(i) \geq 2$	$j := 1..m$
$i := i + 1$	$\text{napló}[i,j] = 5$
	$s := s + 1$ SKIP

Implementáció

Adattípusok megvalósítása

A tervben szereplő mátrixot `vector<vector<int>>`-ként deklaráljuk. Mivel a vektor 0-tól indexelődik, azért a tervbeli ciklusok indextartományai a $0..n-1$ és a $0..m-1$ intervallumra módosulnak, ahol a n -re `t.size()` alakban, m -re pedig `t[i].size()` alakban hivatkozhatunk.

A megvalósításban a diákok és a tantárgyak neveit is tároljuk egy-egy külön tömbben.

Bemenő adatok formája

Az adatokat be lehet olvasni egy szöveges állományból vagy meg lehet adni billentyűzetről. Ha a programot parancssorból indítjuk úgy, hogy paraméterként megadjuk a bemenő adatokat tartalmazó szöveges állomány nevét, akkor innen olvassa be a program az adatokat. Ha nem adunk meg a parancssorban állomány nevet vagy nem parancssorból indul a program, akkor az először megkérdezi az adatbevitel módját, majd a szöveges állományból való olvasást választva bekéri az állomány nevét. A billentyűzetről vezérelt adatbevitelt a program párbeszéd-üzemmódban irányítja, és azt megfelelő adat-ellenőrzésekkel vizsgálja.

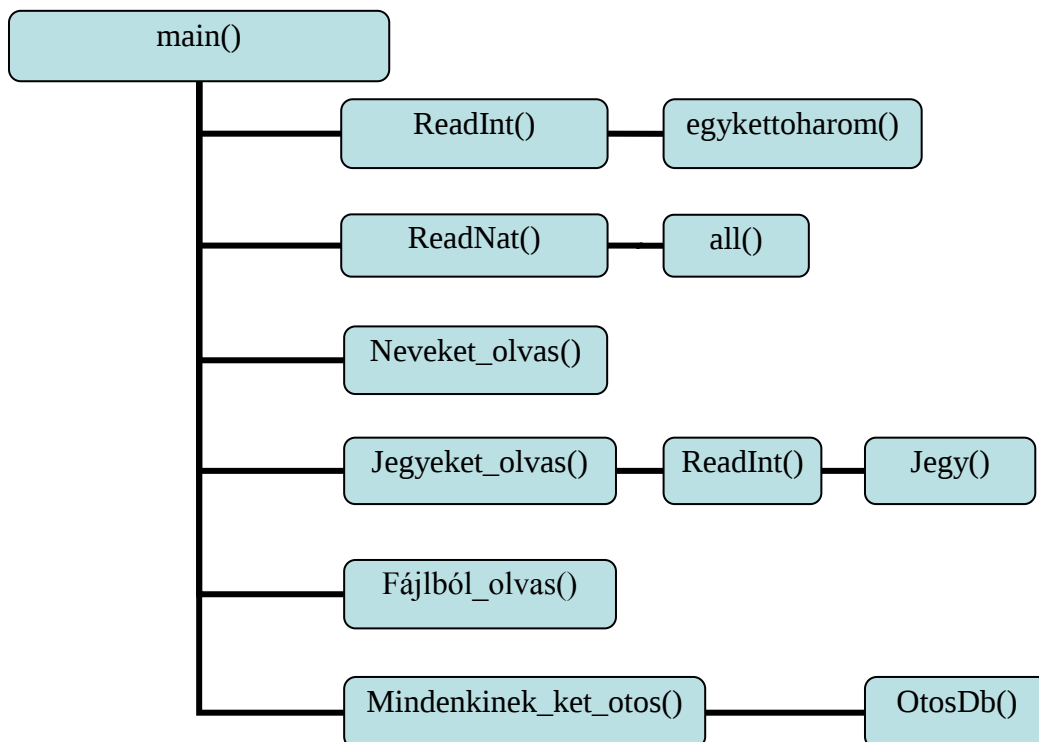
A szöveges állomány formája kötött, arról feltesszük, hogy helyesen van kitöltve, ezért ezt külön nem ellenőrizzük. Az első sor a tanulók és a tantárgyak számát tartalmazza, szóközzel vagy tabulátor jelekkel elválasztva. Ezt követően olvashatók a tanulók nevei soronként, majd a tantárgyak nevei ugyancsak soronként. Végül az osztályzatok következnek a tanulók sorrendjében úgy, hogy minden tanuló jegyei egy sorban szóközzel vagy tabulátor jelekkel legyenek elválasztva a tantárgyak megadott sorrendjében. Minden sor végén (az utolsó sor végén is) sorvége jel legyen.

Példa:

```
2 3
Kerek Berci
Nagy János
Matek
Föci
Orosz
4 3 2
5 4 4
```

Program váz

A program több állományból áll. A *read* csomagban (*read.h*, *read.cpp*) a *ReadInt()*, *ReadNat()*, és az *all()* függvényeket találjuk (ezek az egész számok billentyűzetről való beolvasását támogatják), az összes többi függvény a *naplo.cpp* állományban van.



Tesztelési terv**Tesztesetek a feladat specifikációja alapján (fekete doboz tesztelés)**

Érvényes tesztesetek:

A. Külső programozási tétel (intervallum és lineáris keresés):

1. Üres napló esetei:
(t10.txt: nincs tanuló, sem tantárgy – válasz: igaz)
(t11.txt: nincs tanuló – válasz: igaz)
2. Egy tanuló, egy tantárgy esete. (t2.txt: 5 – válasz: hamis)
3. Több tanuló, több tantárgy: mindenkinek van legalább két ötöse.
(t3.txt: 2×3 [5, 3, 5, 5, 5, 5] – válasz: igaz)
4. Több tanuló, több tantárgy: csak az első tanulónak nincs két ötöse.
(t4.txt: 3×2 [5, 3, 5, 5, 5, 5] – válasz: hamis)
5. Több tanuló, több tantárgy: csak az utolsó tanulónak nincs két ötöse.
(t5.txt: 3×2 [5, 5, 5, 5, 1, 5] – válasz: hamis)
6. Több tanuló, több tantárgy: senkinek nincs két ötöse.
(t6.txt: 3×2 [5, 3, 5, 3, 1, 5] – válasz: hamis)

B. Belső programozási tétel (intervallum és számlálás):

1. Üres napló esetei:
(t10.txt: nincs tanuló, sem tantárgy – válasz: igaz)
(t12.txt: nincs tantárgy – válasz: hamis)
2. Egy tanuló, egy tantárgy esete.
(t2.txt: 5 – válasz: hamis)
3. Egy tanulónak nincs ötöse
(t8.txt: 1×4 [1, 3, 4, 3] – válasz: hamis)
4. Egy tanulónak egyetlen ötöse van
(t9.txt: 1×4 [1, 5, 4, 3] – válasz: hamis)
5. Egy tanulónak két ötöse van
(t13.txt: 1×4 [5, 3, 4, 5] – válasz: igaz)
6. Egy tanulónak sok ötöse van
(t14.txt: 1×4 [5, 5, 5, 3] – válasz: igaz)

Érvénytelen tesztesetek:

1. Nem megengedett értékek:
Nem 1 és 5- közé eső osztályzatok (nem okoz problémát)
Nem számok az osztályzatok (ilyen eseteket eleve kizártunk, nem ellenőrizzük)

Tesztesetek a megoldó kód alapján (fehér doboz tesztelés)

A beolvasást végző függvények tesztelése:

1. Menü választás tesztelése (1, 2, 3, más)
2. Beolvasás mindhárom módozatának tesztelése.
3. Parancssorból indítás fájlnevvvel és anélkül.
4. Nem létező fájlnev megadása.
5. Hibás adatok a billentyűzetről (negatív tanuló vagy tantárgy szám, hibás osztályzat).

A `Mindekinek_ket_otos()` függvény tesztelése:

1. igenlő válasz esete több tanulónál (ciklus feltételnél való kilépés vagy be se lépés esetei):
(t7.txt: 3×2 [5, 5, 5, 5, 5, 5] – válasz: igaz)
(t11.txt: 0×3 [] – válasz: igaz)
2. nemleges válasz, amely az első tanulónál jelentkezik (kilépés az első):
(t6.txt: 3×2 [5, 3, 5, 3, 1, 5] – válasz: hamis)
3. nemleges válasz, amely az utolsó tanulónál jelentkezik (kilépés több menet):
(t5.txt: 3×2 [5, 5, 5, 5, 1, 5] – válasz: hamis)
4. elágazás feltétel tesztelése (kisebb, épp egyenlő, nagyobb, mint 2 esetei):
(t5.txt: 3×2 [5, 5, 5, 5, 1, 5] – válasz: hamis)

A `OtosDb()` függvény tesztelése:

1. olyan adatsorra, amikor egyszer sem lép a ciklusba a vezérlés;
(t12.txt: 2×0 [] – válasz: 0)
2. olyan adatsorra, amikor egyszerem növeljük a darabszámot;
(t8.txt: 1×4 [1, 3, 4, 3] – válasz: 0)
3. olyan adatsorra, amikor egyszer növeljük a darabszámot;
(t9.txt: 1×4 [1, 5, 4, 3] – válasz: 1)
4. olyan adatsorra, amikor többször növeljük a darabszámot;
(t13.txt: 1×4 [5, 3, 4, 5] – válasz: 2)

Alternatív megoldási terv (rekurzív függvénnyel)

$$Uf = (Ef \wedge (I = \forall i \in [1..n] : \text{jótanuló}(i))) = (Ef \wedge (I = \forall_{i=1}^n \text{search}(\text{jótanuló}(i))))$$

ahol $\text{jótanuló}(i) = \exists j \in [1..m] : \text{ötösdb}_i(j) = 2$

ahol $\text{ötösdb}_i : [0..m] \rightarrow \mathbb{N}$

$$\text{ötösdb}_i(0) = 0$$

$$\text{ötösdb}_i(j) = \text{ötösdb}_i(j-1) + \begin{cases} 1 & \text{ha napló}[i, j] = 5 \\ 0 & \text{ha napló}[i, j] \neq 5 \end{cases} \quad (j \geq 1)$$

$l, i := \text{true}, 1$
$I \wedge i \leq n$
$l := \text{jótanuló}(i)$
$i := i + 1$

$l := \text{jótanuló}(i)$
$l, j, s := \text{false}, 1, 0$
$\neg l \wedge j \leq m$
$\text{napló}[i, j] = 5$
$s := s + 1$ SKIP
$l := s = 2$
$j := j + 1$